UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/040,500	12/28/2001	David J. Long	50277-1767	3700

42425	7590	07/05/2006

HICKMAN PALERMO TRUONG & BECKER/ORACLE
2055 GATEWAY PLACE
SUITE 550
SAN JOSE, CA 95110-1089

EXAMINER
DAO, THUY CHAN

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 07/05/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	10/040,500	LONG ET AL.
	Examiner	Art Unit	
	Thuy Dao	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>03</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☐ Responsive to communication(s) filed on <u>20 April 2006</u>.
2a) ☐ This action is **FINAL.** 2b) ☒ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) <u>1-45</u> is/are pending in the application.
    4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) <u>1-45</u> is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.
10) ☒ The drawing(s) filed on <u>28 December 2001</u> is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
    a) ☐ All   b) ☐ Some * c) ☐ None of:
        1. ☐ Certified copies of the priority documents have been received.
        2. ☐ Certified copies of the priority documents have been received in Application No. _____.
        3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date <u>2/6, 3/15, 5/5/06</u>.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____ .

DETAILED ACTION

1. This action is responsive to the amendment filed on April 20, 2006.

2. Claims 1-45 have been examined. Claims 1 and 24 are independent claims.

Response to Amendments

3. Per Applicant's request, claim 24 has been amended.

Information Disclosure Statement

4. The Office acknowledges receipt of the Information Disclosure Statement filed on February 6, March 15, and May 5, 2006. It has been placed in the application file and the information referred to therein has been considered by the examiner.

Response to Arguments

5. The Applicant is thanked for thought-out reply. Applicant's arguments filed on April 20, 2006 have been fully considered.

Priority Date:

The Applicant stated that *"As long as (a) the disclosure of the previous application, to which the later-filed application claims priority, is sufficient to comply with the requirements of the first paragraph of 35 U.S.C. 112, and (b) there is a valid chain of priority claims between the previous application and the later-filed application ..."* (Remarks, page 13, lines 5-8).

The Examiner respectfully disagrees with this statement.

(a) As set forth in previous Office Actions, the present application (10/040,500) is not a continuation of the parent application (09/853,823) as claimed (i.e., totally different specification and figures). The Applicant only argued but failed to show how and where the disclosure of the previous application is sufficient to comply with the requirements of the first paragraph of 35 U.S.C. 112.

(b) There is not a valid chain of priority claims. The priority date of the US Provisional Application Serial No. 60/204,196 (May 12, 2000) is not acknowledged

because the present application's earliest effective filling date was December 28, 2001 and later than 12 months after the date on which the provisional application was filed (May 12, 2000). See MPEP, Appendix L, section 35 USC 119(e), page 24.

Accordingly, the priority date of this application is the filing date December 28, 2001.

Rejections based on the prior art:

Applicant's arguments filed April 20, 2006 (Remarks, pages 13-21) with respect to the rejections of claims 1-45 are persuasive. Therefore, the finality of the Office Action mailed January 17, 2006 has been withdrawn. However, upon further consideration, a new ground of rejection is made in view of "Oracle.TM. Internet File System Developer's Guide", Release 1.1, published in September 2000.

Claim Rejections – 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1-45 are rejected under 35 U.S.C. 102(b) as being anticipated by "Oracle.TM. Internet File System Developer's Guide", Release 1.1, published September 2000 (hereinafter "iFS-Dguide").

Claim 1:

iFS-Dguide discloses *a computer-implemented method for establishing a structure of a data item within a computer system,*

where said data item is an instance of a first class and inherits attributes and methods from said first class (e.g., page 4-14, lines 7-25, Sample Code: Create Document Instances and two specific instances of the InsuranceForm class; pages 4-

10 to 4-12, Sample Code: Create an Instance Class Bean, the InsuranceForm Java.TM. source code file; page 4-13, lines 9-13, creating document instance files to instantiate the document class), *the method comprising the steps of:*

> *creating a category object that is an instance of a category class,, wherein said category object has one or more attributes,* (e.g., page 5-2, lines 1-16, a standard XML parser or a custom parser is an instance of a Java.TM. parser class; pages 5-10 to 5-15, Java.TM. source code of a custom parser; page 5-11, lines 24-30, one or more attributes); *and*

> *associating said data item with said category object without associating said category object with all other instances of said first class thereby causing said data item to be associated with a structure that includes storage for values for said one or more attributes of said category class* (e.g., page 5-6, section How does XML Parsing Work?; page 5-7, lines 7-16, associating said data item with said parser thereby causing said data item to be associated with document objects, folder objects, or a combination of document and folder objects; page 5-2, lines 1-16; page 5-3, lines 3-14; page 5-9: 6-13; page 5-11: 6-8).

Claim 2:

The rejection of base claim 1 is incorporated. iFS-Dguide also discloses *the computer-implemented steps of: receiving data that is designated for a particular attribute of said one or more attributes; determining whether said data conforms to rules associated with said particular attribute; and if said data conforms to said rules, storing said data as a value into said particular attribute* (e.g., page 4-14, lines 10-25, receiving data that is designed for particular attributes such as Name, ClaimNumber, ClaimType; page 3-2, if said data conforms to rules, storing said data into the iFS repository).

Claim 3:

The rejection of intervening claim 2 is incorporated. iFS-Dguide also discloses *the step of receiving, determining, and storing are performed by a method in said*

category class (e.g., page 5-6, lines 16-34, said steps are performed by a method in said parser class).

Claim 4:

The rejection of intervening claim 2 is incorporated. iFS-Dguide also discloses *said rules are data type rules associated with a data type of said particular attribute* (e.g., page 5-7, lines 7-16, said parser determines which type of object to create based on the InputStream or Reader object passed to it).

Claim 5:

The rejection of intervening claim 2 is incorporated. iFS-Dguide also discloses *said rules are software rules* (e.g., page 5-8: section Write the Parser Class, said rules are software rules implemented in the source code of the Parser class).

Claim 6:

The rejection of base claim 1 is incorporated. iFS-Dguide also discloses *storing within a database, objects that define said data item and said category object* (e.g., pages 5-15 and 5-16, Deploy and Register the Parser; page 5-16, lines 9-13, mapping file extension to parser; page 5-16, lines 29-33, PropertyBundle object stores the mappings of said data item and said parser object).

Claim 7:

The rejection of base claim 1 is incorporated. iFS-Dguide also discloses *maintaining an object relational mapping system that indicates a correlation between said data item and data stored in a relational database* (e.g., page 3-2, section How Documents Are Stored in the Repository).

Claim 8:

The rejection of base claim 1 is incorporated. iFS-Dguide also discloses *said category class is a user defined subclass of a parent category class* (e.g., page 5-8, lines 28-33, user defined parser can subclass an existing iFS parser).

Claim 9:

The rejection of base claim 1 is incorporated. iFS-Dguide also discloses *the step of associating said data item with said category object further includes the computer-implemented step of establishing a pointer from said category object to said data item* (e.g., page 5-16, lines 9-13, mapping file extension to parser).

Claim 10:

The rejection of base claim 1 is incorporated. iFS-Dguide also discloses *associating said data item with said category object further includes the computer-implemented step of maintaining a table that includes an entry that indicates that said data item is associated with said category* (e.g., page 5-16, lines 29-33, PropertyBundle object stores the mappings of said data item and said parser object).

Claim 11:

The rejection of intervening claim 10 is incorporated. iFS-Dguide also discloses *maintaining a table further includes the computer-implemented step of maintaining said entry to include a key that identifies said category object and a pointer to said category object* (e.g., page 5-16, lines 29-33, each Property object stores the mapping between a file extension and a parser as a Name/Value pair).

Claim 12:

The rejection of intervening claim 10 is incorporated. iFS-Dguide also discloses *maintaining a table further includes the computer-implemented step of maintaining said table externally to said data item* (e.g., page 5-17, lines 1-10).

Claim 13:

The rejection of intervening claim 10 is incorporated. iFS-Dguide also discloses *of maintaining a table further includes the computer-implemented step of maintaining said table internally to said data item* (e.g., page 5-17, lines 11-25).

Claim 14:

The rejection of base claim 1 is incorporated. iFS-Dguide also discloses *storing into said data item a hash table; locating an entry in said hash table for said category object based on data associated with said category object; locating a pointer to said category object in said entry; and following said pointer to locate said category object* (e.g., page 5-16, lines 8-13, a file with a certain extension will be passed to a associated parser before it is stored in the repository).

Claim 15:

The rejection of intervening claim 14 is incorporated. iFS-Dguide also discloses *receiving data that is designated as a key for locating said entry in said hash table; determining whether said data conforms to rules associated with said key; and if said data conforms to said rules using said data as said key to locate said entry* (e.g., page 5-10, lines 1-18, optional parameter with datatype Hashtable).

Claim 16:

The rejection of base claim 1 is incorporated. iFS-Dguide also discloses *the step of creating a category object further includes the computer-implemented step of maintaining a table that includes an entry that contains a particular attribute of said one or more attributes* (e.g., page 5-10, lines 19-28, a subclass named CUSTOM with the attribute TITLE, be registered on the server with the file extension .cus).

Claim 17:

The rejection of intervening claim 16 is incorporated. iFS-Dguide also discloses *maintaining a table further includes the computer-implemented step of maintaining said entry to include a key that identifies said particular attribute* (e.g., page 5-12, lines 5-13).

Claim 18:

The rejection of intervening claim 16 is incorporated. iFS-Dguide also discloses *of maintaining a table further includes the computer-implemented step of maintaining said table externally to said category object* (e.g., page 5-17, lines 1-10).

Claim 19:

The rejection of intervening claim 16 is incorporated. iFS-Dguide also discloses *maintaining a table further includes the computer-implemented step of maintaining said table internally to said category object* (e.g., page 5-17, lines 11-25).

Claim 20:

The rejection of base claim 1 is incorporated. iFS-Dguide also discloses *creating a category object further includes the computer-implemented of storing into said category object a hash table; and locating an entry in said hash table for a particular attribute of said one or more attributes* (page 5-10, lines 1-18, optional parameter with datatype Hashtable).

Claim 21:

The rejection of intervening claim 20 is incorporated. iFS-Dguide also discloses *receiving data that is designated as a key for locating said entry in said hash table; determining whether said data conforms to rules associated with said key; and if said data conforms to said rules, using said data as said key to locate said entry* (e.g., page 3-12, section Searching for a Document; page 3-13, subsection Specific Search Methods and Attribute-based Search).

Claim 22:

The rejection of base claim 1 is incorporated. iFS-Dguide also discloses *associating said category object with a second data item that is an instance of a second class, without associating said category object with all other instances of said second class wherein said first class is a different class from said second class* (e.g., page 4-13, lines 9-28, Creating Document Instances, second data item is an instance of a second class Application program, wherein said first class can be any Oracle iFS protocol or user interface).

Claim 23:

The rejection of base claim 1 is incorporated. iFS-Dguide also discloses *said category class is a first file type* (e.g., page 5-2, said parser class is a first file type) *and said category object is a first file of said first file type in a file system* (e.g., page 5-2, lines 17-31, parser file type in a Oracle.TM. file system iFS);

wherein said first class is a second file type (e.g., page 4-14, InsuranceForm class is a second file type) *and said data item is a second file of said second file type in a file system* (e.g., page 5-2, Document file type in the Oracle.TM. file system iFS); *and*

wherein the step of associating includes associating said second file with said first file without associating said first file with all other instances of said second file type thereby causing said second file to be associated with said structure in said file system (e.g., page 5-6 section How does XML parsing work?; page 5-7, lines 7-16, associating said data item with said parser thereby causing said data item to be associated with document objects, folder objects, or a combination of document and folder objects).

Claim 24:

iFS-Dguide discloses *a computer-implemented method for establishing a structure of a data item within a computer system,*

where the data item is an instance of a class and inherits attributes and methods form said class (e.g., page 4-14, lines 7-25, Sample Code: Create Document Instances and two specific instances of the InsuranceForm class; pages 4-10 to 4-12, Sample Code: Create an Instance Class Bean, the InsuranceForm Java.TM. source code file; page 4-13, lines 9-13, creating document instance files to instantiate the document class), *the method comprising the steps of:*

creating a first category object that is an instance of a first category class, wherein said first category class has one or more attributes (e.g., page 5-2, lines 1-16, a standard XML parser or a custom parser is an instance of a Java.TM. parser class; pages 5-10 to 5-15, Java.TM. source code of a custom parser; page 5-11, lines 24-30, one or more attributes);

creating a second category object that is an instance of a second category class, wherein said second category class has one or more attributes and is a different class than the first category class (e.g., page 5-3: 6-14, a second category object is an instance of a second category class ClassSelectionParser; page 5-4: 5-11, second category class ClassSelectionParser has one or more attributes and is a different class than the first category standard/custom parser class);

wherein the first category class and the second category class are external to the class lineage of the class of which the data item is an instance (e.g., standard/custom parser classes and ClassSelectionParser class are external to InsuranceForm class); *and*

associating said data item with said first category object and with said second category object thereby causing said data item to be associated with a structure that includes storage for values for said one or more attributes of said first category object and for said one or more attributes of said second category object (e.g., page 5-3: 6-14; page 5-4, line 17 – page 5-5, line 40).

Claims 25-44:

The rejection of base claim 24 is incorporated. Claims 25-44 recite the same limitations as those of the claims 2-21, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of the above claims, they also teach all of the limitations of claims 25-44.

Claim 45:

The rejection of base claim 24 is incorporated. iFS-Dguide also discloses *said first category class is a first file type and said first category object is a first file of said first file type in a file system* (e.g., page 5-3: 3-14, SimpleXmlParser or a custom parser is a first file type, first category object is a first file in a Oracle.TM. iFS file system);·

wherein said second category class is a second file type and said second category object is a second file of said second file type in said file system (e.g., page 5-3: 3-14, ClassSelectionParser is a second file type and said second category object is a second file of said iFS file system);

wherein said class is a third file type and said data item is a third file of said third file type in said file system (e.g., InsuranceForm class is a third file type and said data item is a third file of said iFS file system); *and*

wherein the step of associating includes associating said third file with said first file and said second file thereby causing said third file to be associated with said structure in said file system that includes storage for values for said one or more attributes of said first file type and for said one or more attributes of said second file type (e.g., page 5-3: 6-14; page 5-4, line 17 – page 5-5, line 40).

8. Claims 1 and 24 are rejected under 35 U.S.C. 102(b) as being anticipated by iFS-Dguide.

Claim 1:

iFS-Dguide discloses *a computer-implemented method for establishing a structure of a data item within a computer system* (e.g., page 3-2: 1-10),

where said data item is an instance of a first class and inherits attributes and methods from said first class (e.g., page 3-6: 7-8, "Hello_World.txt" is an instance of the Java string class; page 3-10: 1-17), *the method comprising the steps of:*

creating a category object that is an instance of a category class (e.g., page 3-6: 6, newDocDef object is an instance of a category class DocumentDefinition; page 3-5: 1-24),

wherein said category object has one or more attributes (e.g., page 2-12: 16-31, class DocumentDefinition is a subclass of class PublicObjectDefinition and has one or more attributes); *and*

associating said data item with said category object without associating said category object with all other instances of said first class (e.g., page 3-6: 7-8, said data item as "Hello_World.txt" and said category object as newDocDef:

```
newDocDef.setAttribute("Name",

    AttributeValue.newAttributeValue("Hello_World.txt"));
```

thereby causing said data item to be associated with a structure that includes storage for values for said one or more attributes of said category class (e.g., page 3-9: 1-11 and 20-28; page 3-2: 1-10).

Claim 24:

iFS-Dguide discloses *a computer-implemented method for establishing a structure of a data item within a computer system* (e.g., page 3-2: 1-10),

where the data item is an instance of a class and inherits attributes and methods form said class (e.g., page 3-6: 7-8, "Hello_World.txt" is an instance of the Java string class; page 3-10: 1-17), *the method comprising the steps of:*

creating a first category object that is an instance of a first category class (e.g., page 3-6: 6, newDocDef object is an instance of a first category class DocumentDefinition; page 3-5: 1-24),

wherein said category object has one or more attributes (e.g., page 2-12: 16-31, class DocumentDefinition is a subclass of class PublicObjectDefinition and has one or more attributes);

creating a second category object that is an instance of a second category class (e.g., page 3-6: 16, doc is an instance of a second category class Document),

wherein said second category class has one or more attributes (e.g., page 2-12: 1-15, Document class is a subclass of class PublicObject and has one or more attributes) *and is a different class than the first category class* (e.g., second category class Document is different with the first category class DocumentDefinition);

wherein the first category class and the second category class are external to the class lineage of the class of which the data item is an instance (e.g., DocumentDefinition class and Document class are external to the class lineage of said data item as "Hello_World.txt", an instance of said Java string class); *and*

associating said data item with said first category object and with said second category object (e.g., page 3-6: 5-10 and 11-22)

thereby causing said data item to be associated with a structure that includes storage for values for said one or more attributes of said first category object and for said one or more attributes of said second category object (e.g., page 3-9: 1-11 and 20-28; page 3-6: 11-22).

Conclusion

10. Any inquiry concerning this communication should be directed to examiner Thuy Dao (Twee), whose telephone is (571) 272 8570. The examiner can normally be reached on Monday – Friday from 6:30AM to 3:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

T. Dao

TUAN DAM
SUPERVISORY PATENT EXAMINER